



Try the *new* Portal design

Give us your opinion after using it.

## Search Results

Search Results for: [DEF<AND>((macro<AND>(((string))) )) ]  
Found **187** of **129,763** searched.

## Search within Results



[> Advanced Search](#) [> Search Help/Tips](#)

Sort by: Title Publication Publication Date Score Binder

Results 1 - 20 of 187 short listing



1 2 3 4 5 6 7 8 9 10



- 1 Technical correspondence: Using smgn for rapid prototyping of small domain-specific languages 88%

Holger M. Kienle

**ACM SIGPLAN Notices** September 2001

Volume 36 Issue 9

This paper presents smgn, a grammar-based tool that provides support for scanning, parsing, and automatic parse tree construction. The parse tree can be easily navigated and manipulated with a specific macro language while conveniently generating textual output. smgn is easy to learn and well suited for rapid prototyping of small domain-specific languages. It is part of the SUIF compiler system, where it has been used for the rapid development of the *Hoof* domain-specific language. Further ...

- 2 APAREL—A parse-request language 88%

R. M. Balzer , D. J. Farber

**Communications of the ACM** November 1969

Volume 12 Issue 11

APAREL is described: this language is an extension to an algorithmic language (PL/I) that provides the pattern-matching capabilities normally found only in special purpose languages such as SNOBOL4 and TMG. This capability is provided through parse-requests stated in a BNF-like format. These parse-requests form their own programming language with special sequencing rules. Upon successfully completing a parse-request, an associated piece of PL/I code is executed. This code has available for ...

- 3 Microbe: A self commenting microassembler 88%

Ben A. Laws

**Proceedings of the 10th annual workshop n Micropr gramming** October 1977

A microassembler is described that generates a symbolic commentary for each microinstruction. This approach yields many of the benefits of programming in a high level language without the high implementation cost. The commentary assists with code

optimization and provides a record of the processor's operation. An outline for implementing a commentator gives a sense for its cost, which is small compared to the returned value.

#### 4 ASAP—a simple assertion pre-processor

88%



Igor D.D. Curcio

**ACM SIGPLAN Notices** December 1998

Volume 33 Issue 12

Assertions are widely known as a powerful tool to detect software faults during the debugging of software systems. Despite the maturity of software engineering tools, assertions are seldom used in practice. ASAP is a pre-processor for C programs which implements several concepts defined in the theory of formal specification, such as preconditions, postconditions, assertions related to intermediate states, loop invariants and variants, existential and universal quantifiers. In this paper, the noti ...

#### 5 Generation of Compiler Symbol Processing Mechanisms from Specifications

87%



Stephen P. Reiss

**ACM Transactions on Programming Languages and Systems (TOPLAS)** April 1983

Volume 5 Issue 2

#### 6 Computational parallels between the regular and context-free languages

85%



H. B. Hunt , D. J. Rosenkrantz

**Proceedings of the sixth annual ACM symposium on Theory of computing** April 1974

This paper presents a complexity theory of formal languages. The main technique used is that of embedding " $\equiv_{0,1}^*$ ", " $\equiv_{0,1}$ ", and " $\equiv_{0,1}^f$ " into other linguistic predicates. In Section 2, the undecidability of " $\equiv_{0,1}^*$ " for cfl's is exploited to provide sufficient conditions for the undecidability of predicates on the cfl's. In Section 3, the same techniques are applied to regular sets. Predicates satisfying conditions sim ...

#### 7 On the complexity of grammar and related problems

85%



H. B. Hunt , T. G. Szymanski

**Proceedings of seventh annual ACM symposium on Theory of computing** May 1975

In [1] and [2] a complexity theory for formal languages and automata was developed. This theory implies most of the previously known results and yields many new results as well. Here we develop an analogous theory for several classes of more practically motivated problems. Two such classes, both closely related to formal language and automata theory, suggest themselves - grammar problems and program scheme problems. Here, our primary emphasis is on grammar problems of interest in parsing an ...

#### 8 Cascade compilation revisited

84%



Ted Stern , Daniel Grimwood

**ACM SIGPLAN Fortran Forum** April 2002

Volume 21 Issue 1

The problem of cascade recompilation in large modern Fortran projects is well known --- a non-interface related change of a module can cause recompilation in the entire chain of dependents of that module, even when the compiler would not require it [1]. Furthermore, it is not trivial to set up the proper dependency relationship among modules and their dependents. See [2-5] for examples of various ways of setting up these dependencies. Solutions proposed up until now have not dealt comprehensively ...

#### 9 Revised5 report on the algorithmic language scheme

84%



N. I. Adams , D. H. Bartley , G. Brooks , R. K. Dybvig , D. P. Friedman , R. Halstead , C.

Hanson , C. T. Haynes , E. Kohlbecker , D. Oxley , K. M. Pitman , G. J. Rozas , G. L. Steele , G.

**10 Generative communication in Linda**

84%



David Gelernter

**ACM Transactions on Programming Languages and Systems (TOPLAS)** January 1985  
Volume 7 Issue 1

Generative communication is the basis of a new distributed programming language that is intended for systems programming in distributed settings generally and on integrated network computers in particular. It differs from previous interprocess communication models in specifying that messages be added in tuple-structured form to the computation environment, where they exist as named, independent entities until some process chooses to receive them. Generative communication results in a number ...

**11 Fast detection of communication patterns in distributed executions**

83%



Thomas Kunz , Michiel F. H. Seuren

**Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research** November 1997

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

**12 Joining separate file systems**

83%



Jon Livesey

**Proceedings of the 20th annual Southeast regional conference** April 1982

This paper takes a look at the so-called 'meta-system' approach to Distributed System construction; that is, constructing a distributed system by tying together existing local operating systems. In particular it looks at some of the file system problems that may be encountered. These problems are simplified if one has the opportunity to rewrite the underlying local operating system (see [Oppen 81], for example), but typically, this is not the case. Since it is impractical to look at all existin ...

**13 A rapid method for digital filtering**

83%



John R. B. Whittlesey

**Communications of the ACM** September 1964  
Volume 7 Issue 9

Since much of the computer time spent in time-series analysis is used for multiplications, a minimum multiplication method was devised for digital filtering, with the expectation that it would be useful in the online, real-time analysis of biological data. The filters are constructed from a succession of readily analyzable components in a manner that facilitates cascading. The repertoire of frequency response curves includes relatively good low-pass and band-pass designs. Programs are avail ...

**14 EPIC - a retargetable, highly optimizing Lisp compiler**

82%



R. R. Kessler , J. C. Peterson , H. Carr , G. P. Duggan , J. Knell


**ACM SIGPLAN Notices , Proceedings of the 1986 SIGPLAN symposium on Compiler construction** July 1986  
Volume 21 Issue 7

The Experimental Portable Standard Lisp Compiler (EPIC) is a compiler testbed for

experimentation with, and development of, Lisp compilation strategies. EPIC uses an architectural description of the target machine to increase portability, and performs extensive optimizations in the form of source-to-source transformations, register allocation, and peephole optimization. It introduces machine-specific instructions early to enable machine-specific optimizations in the initial passes. EPIC pro ...


## 15 Common Lisp Object System specification

82%

 Daniel G. Bobrow , Linda G. DeMichiel , Richard P. Gabriel , Sonya E. Keene , Gregor Kiczales , David A. Moon  
**ACM SIGPLAN Notices** September 1988  
Volume 23 Issue SI

## 16 Machine language programming in an undergraduate computer science

82%

 curriculum  
Ian Sommerville  
**ACM SIGCSE Bulletin , Proceedings of the seventh SIGCSE technical symposium on Computer science education** February 1977  
Volume 9 Issue 1

This paper examines the advantages and disadvantages of teaching machine language programming to computer science undergraduate students. A teaching language based on reverse Polish notation, but with high-level control constructs, is presented as an alternative to conventional assembly language. Experiences with using this language are described.

## 17 A LISP compiler producing compact code


82%

 William Rowan  
**Proceedings of the 1980 ACM conference on LISP and functional programming** August 1980

A compiler has been written which compiles MACLISP into a compact intermediate language called 1-code, and an 1-code interpreter has been incorporated into an existing LISP system. The 1-code "machine" has a simple stack architecture and an instruction set specifically designed for LISP. Compiled programs consist of a string of eight-bit bytes of 1-code, and a local table of quantities used by the compiled code. The system has been used to compile most of the MACSYMA ...


## 18 Document Formatting Systems: Survey, Concepts, and Issues

82%

 Richard Furuta , Jeffrey Scofield , Alan Shaw  
**ACM Computing Surveys (CSUR)** September 1982  
Volume 14 Issue 3

## 19 Toward a logical/physical theory of spreadsheet modeling


82%

 Tomás Isakowitz , Shimon Schocken , Henry C. Lucas  
**ACM Transactions on Information Systems (TOIS)** January 1995  
Volume 13 Issue 1

In spite of the increasing sophistication and power of commercial spreadsheet packages, we still lack a formal theory or a methodology to support the construction and maintenance of spreadsheet models. Using a dual logical/physical perspective, we identify four principal components that characterize any spread sheet model: schema, data, editorial, and binding. We present a factoring algorithm for identifying and extracting these components ...

## 20 The Pan language-based editing system

82%

 Robert A. Ballance , Susan L. Graham , Michael L. Van De Vanter

**ACM Transactions on Software Engineering and Methodology (TOSEM)** January 1992  
Volume 1 Issue 1

Powerful editing systems for developing complex software documents are difficult to engineer. Besides requiring efficient incremental algorithms and complex data structures, such editors must accommodate flexible editing styles, provide a consistent, coherent, and powerful user interface, support individual variations and projectwide configurations, maintain a sharable database of information concerning the documents being edited, and integrate smoothly with the other tools in the environme ...

---

**Results 1 - 20 of 187**

**short listing**

  
Prev  
Page

**1 2 3 4 5 6 7 8 9 10**

  
Next  
Page

---

The ACM Portal is published by the Association for Computing Machinery. Copyright ? 2004 ACM, Inc.

## Welcome to IEEE Xplore®

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

## Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

## Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced

## Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

Your search matched **4** of **1015452** documents.

A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

**Refine This Search:**

You may refine your search by editing the current search expression or entering a new one in the text box.


☐ Check to search within this result set
**Results Key:**

**JNL** = Journal or Magazine   **CNF** = Conference   **STD** = Standard

**1 Hybrid genetic algorithms for constrained placement problems**

*Schnecke, V.; Vornberger, O.;*

Evolutionary Computation, IEEE Transactions on , Volume: 1 , Issue: 4 , Nov. 1997

Pages:266 - 277

[\[Abstract\]](#)   [\[PDF Full-Text \(280 KB\)\]](#)   IEEE JNL

**2 An instrument control and data analysis program configured for NMR imaging**

*Roos, M.S.; Mushlin, R.A.; Veklerov, E.; Port, J.D.; Ladd, C.; Harrison, C.G.;*

Nuclear Science, IEEE Transactions on , Volume: 36 , Issue: 1 , Feb. 1989

Pages:988 - 992

[\[Abstract\]](#)   [\[PDF Full-Text \(476 KB\)\]](#)   IEEE JNL

**3 CPP denotational semantics**

*Favre, J.-M.;*

Source Code Analysis and Manipulation, 2003. Proceedings. Third IEEE International Workshop on , 26-27 Sept. 2003

Pages:22 - 31

[\[Abstract\]](#)   [\[PDF Full-Text \(405 KB\)\]](#)   IEEE CNF

**4 Splicing systems on graphs**

*Freund, R.;*

Intelligence in Neural and Biological Systems, 1995. INBS'95, Proceedings., First International Symposium on , 29-31 May 1995

Pages:189 - 194

[\[Abstract\]](#)   [\[PDF Full-Text \(408 KB\)\]](#)   IEEE CNF

---

[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#) | [Search by Author](#) | [Basic Search](#) | [Advanced Search](#) | [Join IEEE](#) | [Web Account](#) | [New this week](#) | [OPAC Linking Information](#) | [Your Feedback](#) | [Technical Support](#) | [Email Alerting](#) | [No Robots Please](#) | [Release Notes](#) | [IEEE Online Publications](#) | [Help](#) | [FAQ](#) | [Terms](#) | [Back to Top](#)

Copyright © 2004 IEEE — All rights reserved

Searching for **string and macro**.

Restrict to: [Header](#) [Title](#) Order by: [Expected citations](#) [Hubs](#) [Usage](#) [Date](#) Try: [Amazon](#) [B&N](#) [Google \(RI\)](#) [Google \(Web\)](#) [CSB](#) [DBLP](#)

468 documents found. Only retrieving 125 documents (System busy - maximum reduced). Order: number of citations.

[Query-Limited Reducibilities - Beigel \(1995\)](#) (Correct) (28 citations)

oracle is associated with a particular set A of **strings** (or natural numbers) the oracle is said to be the set A. When the computer needs to know if the **string** (or natural number) x belongs to A, the computer was typeset using Leslie Lamport's L A T E X **macro** package for Donald E. Knuth's T E X typesetting [www.eecs.uic.edu/~beigel/papers/dissertation.PS.gz](http://www.eecs.uic.edu/~beigel/papers/dissertation.PS.gz)

[Programmable Syntax Macros - Weise, Crew \(1993\)](#) (Correct) (26 citations)

and return, advanced control statements, and data-structuring facilities. The additional programming to transformations on trees, rather than on token **streams**. His **macro** system was substitution based, that Programmable Syntax Macros Daniel Weise Roger Crew Microsoft Research [seclab.cs.ucdavis.edu/~devanbu/teaching/260/macros.pdf](http://seclab.cs.ucdavis.edu/~devanbu/teaching/260/macros.pdf)

[FunnelWeb User's Manual - Ross Williams For \(1992\)](#) (Correct) (12 citations)

.34 1.7.3 Hierarchical Structure .34 1.7.4  
44 2.6 Changing the Strength of Headings .46 2.7  
[www.loria.fr/services/tex/english/./litte/u\\_manual.pdf](http://www.loria.fr/services/tex/english/./litte/u_manual.pdf)

[Multiple Default Inheritance in a Unification-Based - Graham Russell John \(1991\)](#) (Correct) (10 citations)

Russell, G.J. Carroll and S. Warwick (1991) Multiple default inheritance in [acl.ldc.upenn.edu/P/P91/P91-1028.pdf](http://acl.ldc.upenn.edu/P/P91/P91-1028.pdf)

[Transducers from Rewrite Rules with Backreferences - Dale Gerdemann University \(1999\)](#) (Correct) (5 citations)

macs uses the special brackets (and )to capture **strings** along with the notation n to recall the nth such along with the notation n to recall the nth such **string**. The expression (a\*bl matches **strings** of the Table 1: Regular expression operators. **macro** (vowel, a, e, i, o, u)ln such **macro** [acl.ldc.upenn.edu/E/E99/E99-1017.pdf](http://acl.ldc.upenn.edu/E/E99/E99-1017.pdf)

[XP: A Common Lisp Pretty Printing System - Waters \(1989\)](#) (Correct) (5 citations)

[publications.ai.mit.edu/ai-publications/1000-1499/AIM-1102a.ps](http://publications.ai.mit.edu/ai-publications/1000-1499/AIM-1102a.ps)

[U.s. Department Of Commerce - National Institute Of \(1993\)](#) (Correct) (5 citations)

2 Structure of the  
4.1.10  
[www.nist.gov/msidlibrary/doc/clark92.ps](http://www.nist.gov/msidlibrary/doc/clark92.ps)

[OpenJava: A Class-based Macro System for Java - Tatsubori, Chiba, Killijian.. \(2000\)](#) (Correct) (4 citations)

since they require the ability to access a logical structure of programs. One of the drawbacks of system for not only behavioral reflection but also structural reflection. A key idea of our **macro** OpenJava: A Class-Based Macro System for Java Michiaki Tatsubori 1 Shigeru [www.csg.is.titech.ac.jp/~mich/pub/200007\\_lncs1826.pdf](http://www.csg.is.titech.ac.jp/~mich/pub/200007_lncs1826.pdf)

[A Finite State and Data-Oriented Method for Grapheme.. - Gosse Bouma.. \(2000\)](#) (Correct) (3 citations)

[Bouma, 2000] Bouma, G. 2000)A finite-state and data-oriented method for grapheme to phoneme [acl.ldc.upenn.edu/A/A00/A00-2040.pdf](http://acl.ldc.upenn.edu/A/A00/A00-2040.pdf)

[The Effects of Tag-Mediated Selection of Partners in Evolving.. - Riolo \(1996\)](#) (Correct) (3 citations)

for agents playing the IPD is described. The agents strategies are represented as (i q q) triples as in [Axelrod and Dion, 1988] for a review)The IPD game structure captures much of the tension between the graphically and, more generally, the micro- and **macr** -dynamics are easier to understand. The key [www.pscs.umich.edu/SCS/reprints/ipd1.ps.gz](http://www.pscs.umich.edu/SCS/reprints/ipd1.ps.gz)



FIPA Communicative Act Library Specification - Document Title Fipa (2001) (Correct) (3 citations)  
messages are modified compatible with new ACL **string** representation and SL specs. 2000/10/19 Edited  
permission from the holder(s) of the rights. FIPA strongly encourages anyone implementing any part of  
by providing a standard set of composite and **macro** communicative acts, derived from the FIPA  
liawww.epfl.ch/~iagents/FIPASpecs/pc00037e.prn.pdf

Literate-Programming Can Be Simple and Extensible - Ramsey (1993) (Correct) (2 citations)  
of Computer Science, Princeton University 35 Olden Street, Princeton, New Jersey 08544 October 1993  
provides an integer representation for **string** literals, and implements a simple form of  
helps explain the program. The code parts are like **macro** definitions they have names, and they contain  
www.loria.fr/services/tex/english/./litte/ieee.pdf

HELM and the Semantic Math-Web - Asperti, Padovani, Coen, Schena (Correct) (2 citations)  
of XML technology to the large repositories of structured, content oriented information oered by  
suitable for the creation of large repositories of structured mathematical knowledge accessible via Web.  
a Constructor is an arbitrary term (an ENTITY is a **macro** declaration)noParams, 3 DTDs are going to be  
www.cs.unibo.it/~sacerdot/smweb.ps.gz

FIPA SL Content Language Specification - Document Title Fipa (2000) (Correct) (2 citations)  
permission from the holder(s) of the rights. FIPA strongly encourages anyone implementing any part of  
Parameter\* Constant =NumericalConstant |String |DateTime. NumericalConstant =Integer |Float.  
Referential operator and is used in the inform-ref **macro** act and other CAs derived from it. Other valid  
liawww.epfl.ch/~iagents/FIPASpecs/x00008f.prn.pdf

Retargetable Functional Simulator - Chandra (1999) (Correct) (2 citations)  
2 Overview of Integrated Environment 7 2.1 Overall Structure .  
.12 2.4.1 Structure of IR .  
and branch prediction. We have also developed a **Macro** Preprocessor (nMP) for processing SimnML **macros**.  
www.cse.iitk.ac.in/pub/moona/sim-nml/simnml-fsim-thesis.ps.gz

Document Structure and Multilingual Authoring - Caroline Brun Marc (2000) (Correct) (1 citation)  
Document Structure and Multilingual Authoring Caroline Brun  
docu- mentation. An XML document is a mixture of struc- ture (the tags) and surface (text between the  
to play a prominent role, as opposed to simply its **macro**-structure (its organization in large semantic  
acl ldc.upenn.edu/W/W00/W00-1404.pdf

SISC: A Complete Scheme Interpreter in Java - Miller (2002) (Correct) (1 citation)  
sisc.sourceforge.net/sisc.ps.gz

Phobos: A front-end approach to extensible compilers - Granicz, Hickey (2002) (Correct) (1 citation)  
13. A. Granicz and J. Hickey. Phobos: A front-end approach to  
mojave.cs.caltech.edu/papers/hicss-36-tr.ps

The Effects and Evolution of Tag-Mediated Selection of Partners in ... - Riolo (1997) (Correct) (1 citation)  
[Axelrod and Dion, 1988] for a review)The IPD game structure captures much of the tension between the  
in the game. Individuals' choices (i.e.their **strategies**) are assumed to be a function of some  
graphically and, more generally, the micro- and **macro**-dynamics are easier to understand. Experiments  
www.pscs.umich.edu/CSCS/reprints/icga97-ipd.ps

*First 20 documents* [Next 20](#)

Try your query at: [Amazon](#) [Barnes & Noble](#) [Google \(RI\)](#) [Google \(Web\)](#) [CSB](#) [DBLP](#)

CiteSeer.IST - Copyright [NEC](#) and [IST](#)